**Provide NFS network shares to specific clients.**

In this tutorial, the **NFS** server is called **nfsserver.example.com** and the **NFS** client **nfsclient.example.com**.

**NFS Server Configuration**

Install the **file-server** package group:

# **yum groupinstall -y file-server**

Add a new service to the firewall:

# **firewall-cmd --permanent --add-service=nfs**

success

Note: **NFSv4** is the version used at the exam and doesn’t need any extra firewall configuration. However, beyond the exam objectives, if you plan to use **NFSv3**, you will also need to run these commands:

# **firewall-cmd --permanent --add-service=mountd**

# **firewall-cmd --permanent --add-service=rpc-bind**

Reload the firewall configuration:

# **firewall-cmd --reload**

success

Activate the **NFS** services at boot:

# **systemctl enable rpcbind nfs-server**

Note: The **nfs-idmap/nfs-idmapd** (changes happened with **RHEL 7.1**) and **nfs-lock** services are automatically started by the **nfs-server** service. **nfs-idmap**/**nfs-idmapd** is required by **NFSv4** but doesn’t allow you any **UID/GID mismatches** between clients and server. It is only used when setting **ACL** by names or to display user/group names.  
All permission checks are still done with the **UID/GID** used by the **server** (see this [thread about nfs-idmap](http://thread.gmane.org/gmane.linux.nfsv4/7103/focus=7105) for more details).

Start the **NFS** services:

# **systemctl start rpcbind** **nfs-server**

Note1: By default, **8 NFS threads** are used (**RPCNFSDCOUNT=8** in the **/etc/sysconfig/nfs** file). This should be increased in a production environment to at least **32** (source: <http://initrd.org/wiki/NFS_Setup>).  
Note2: Optionally, to enable **SELinux** **Labeled NFS Support**, edit the **/etc/sysconfig/nfs** file and paste the following line ([source](https://access.redhat.com/documentation/en-US/Red_Hat_Enterprise_Linux/7/html/SELinux_Users_and_Administrators_Guide/sect-Managing_Confined_Services-NFS-Configuration_Examples.html)): **RPCNFSDARGS=”-V 4.2″**

Create directories to export and assign access rights:

# **mkdir -p /home/tools**

# **chmod 0777 /home/tools**

# **mkdir -p /home/guests**

# **chmod 0777 /home/guests**

Assign the correct **SELinux** contexts to the new directories:

# **yum install -y setroubleshoot-server**

# **semanage fcontext -a -t public\_content\_rw\_t "/home/tools(/.\*)?"**

# **semanage fcontext -a -t public\_content\_rw\_t "/home/guests(/.\*)?"**

# **restorecon -R /home/tools**

# **restorecon -R /home/guests**

Note: The **public\_content\_rw\_t** context is not the only available, you can also use the **public\_content\_ro\_t** (only read-only) or **nfs\_t** (more limited) contexts according to your needs.

Check the **SELinux** booleans used for **NFS**:

# **semanage boolean -l | egrep "nfs|SELinux"**

SELinux boolean State Default Description

xen\_use\_nfs (off , off) Allow xen to use nfs

virt\_use\_nfs (off , off) Allow virt to use nfs

mpd\_use\_nfs (off , off) Allow mpd to use nfs

nfsd\_anon\_write (off , off) Allow nfsd to anon write

ksmtuned\_use\_nfs (off , off) Allow ksmtuned to use nfs

git\_system\_use\_nfs (off , off) Allow git to system use nfs

virt\_sandbox\_use\_nfs (off , off) Allow virt to sandbox use nfs

logrotate\_use\_nfs (off , off) Allow logrotate to use nfs

git\_cgi\_use\_nfs (off , off) Allow git to cgi use nfs

cobbler\_use\_nfs (off , off) Allow cobbler to use nfs

httpd\_use\_nfs (off , off) Allow httpd to use nfs

sge\_use\_nfs (off , off) Allow sge to use nfs

ftpd\_use\_nfs (off , off) Allow ftpd to use nfs

sanlock\_use\_nfs (off , off) Allow sanlock to use nfs

samba\_share\_nfs (off , off) Allow samba to share nfs

openshift\_use\_nfs (off , off) Allow openshift to use nfs

polipo\_use\_nfs (off , off) Allow polipo to use nfs

use\_nfs\_home\_dirs (off , off) Allow use to nfs home dirs

nfs\_export\_all\_rw (on , on) Allow nfs to export all rw

nfs\_export\_all\_ro (on , on) Allow nfs to export all ro

Note1: The **State** column respectively shows the **current** boolean configuration and the **Default** column the **permanent** boolean configuration.  
Note2: Here we are interested in the **nfs\_export\_all\_rw**, **nfs\_export\_all\_ro** and potentially **use\_nfs\_home\_dirs** booleans.  
Note3: The **nfs\_export\_all\_ro** boolean allows files to be shared through **NFS** in **read-only** mode but doesn’t restrict them from being used in **read-write** mode. It’s the role of the **nfs\_export\_all\_rw** boolean to allow **read-write** mode.

If necessary, assign the correct setting to the **SELinux** booleans:

# **setsebool -P nfs\_export\_all\_rw on**

# **setsebool -P nfs\_export\_all\_ro on**

# **setsebool -P use\_nfs\_home\_dirs on**

Edit the **/etc/exports** file and add the following lines with the name (or IP address) of the client(s):

**/home/tools nfsclient.example.com(rw,no\_root\_squash)**

**/home/guests nfsclient.example.com(rw,no\_root\_squash)**

Note: Please, don’t put any space before the opening parenthesis, this would completely change the meaning of the line!

Export the directories:

# **exportfs -avr**

exporting nfsclient.example.com:/home/guests

exporting nfsclient.example.com:/home/tools

# **systemctl restart nfs-server**

Note: This last command shouldn’t be necessary in the future. But, for the time being, it avoids rebooting.

Check your configuration:

# **showmount -e localhost**

Export list for localhost:

/home/guests nfsclient.example.com

/home/tools nfsclient.example.com

Note: You can test what is exported by the **NFS** server from a remote client with the command **showmount -e nfsserver.example.com** but you first need to stop **Firewalld** on the **NFS** server (or open the **111 udp** and **20048 tcp** ports on the **NFS** server).

**NFS Client Configuration**

On the client side, the commands are:

# **yum install -y nfs-utils**

# **mount -t nfs nfsserver.example.com:/home/tools /mnt**